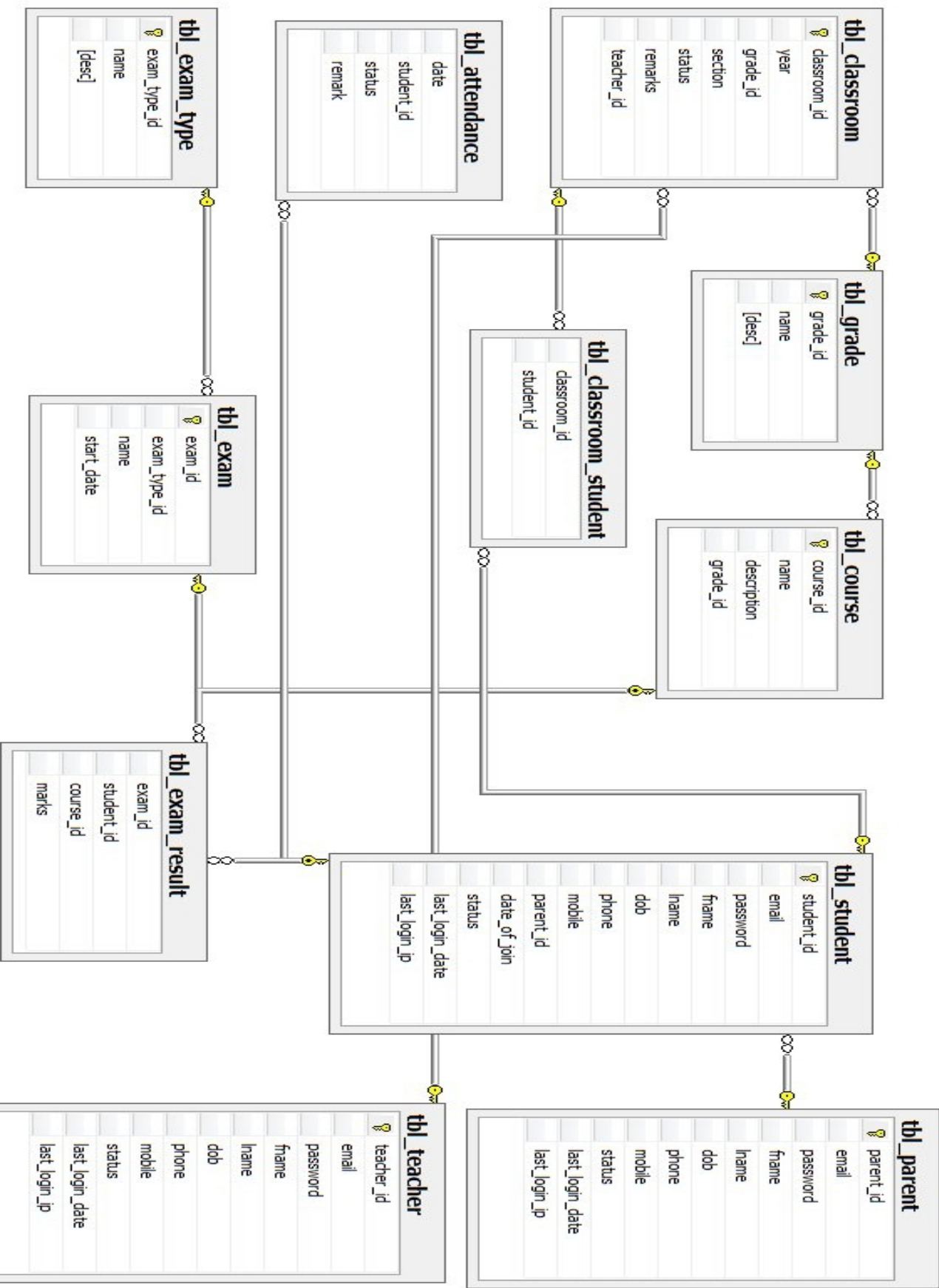


**Mohamed Magdy Sallam**

**Section 3**

**Database Section Report 1**



## Database Description:

This is a school database that contains data about the students' name, date of birth, phone and mobile numbers, their website activity such as last time logged in and last IP. It also contains the same data for the parents and teachers.

Also, the database contains data for the attendance, exams, classrooms and other related stuff as shown in the database diagram.

## Database Creation Code:

```
1      -- CREATE DATABASE db_students          "apply it separately"
2
3      USE db_students
4
5      -- parent table
6      CREATE TABLE tbl_parent (
7      parent_id INT NOT NULL PRIMARY KEY,
8      email VARCHAR(45),
9      password VARCHAR(45),
10     fname VARCHAR(45),
11     lname VARCHAR(45),
12     dob DATE,
13     phone VARCHAR(15),
14     mobile VARCHAR(15),
15     status BIT,
16     last_login_date DATE,
17     last_login_ip VARCHAR(45)
18     );
19
20     -- teacher table
21     CREATE TABLE tbl_teacher (
22     teacher_id INT NOT NULL PRIMARY KEY,
23     email VARCHAR(45),
24     password VARCHAR(45),
25     fname VARCHAR(45),
26     lname VARCHAR(45),
27     dob DATE,
28     phone VARCHAR(15),
29     mobile VARCHAR(15),
30     status BIT,
31     last_login_date DATE,
32     last_login_ip VARCHAR(45)
33     );
34
35
```

```

36      -- student table
37      CREATE TABLE tbl_student (
38      student_id INT NOT NULL PRIMARY KEY,
39      email VARCHAR(45),
40      password VARCHAR(45),
41      fname VARCHAR(45),
42      lname VARCHAR(45),
43      dob DATE,
44      phone VARCHAR(15),
45      mobile VARCHAR(15),
46      parent_id INT FOREIGN KEY REFERENCES tbl_parent(parent_id),
47      date_of_join DATE,
48      status BIT,
49      last_login_date DATE,
50      last_login_ip VARCHAR(45)
51      );
52
53      -- grade table
54      CREATE TABLE tbl_grade (
55      grade_id INT NOT NULL PRIMARY KEY,
56      name VARCHAR(45),
57      [desc] VARCHAR(45)
58      );
59
60      -- exam_type table
61      CREATE TABLE tbl_exam_type (
62      exam_type_id INT NOT NULL PRIMARY KEY,
63      name VARCHAR(45),
64      [desc] VARCHAR(45)
65      );
66
67      -- exam table
68      CREATE TABLE tbl_exam (
69      exam_id INT NOT NULL PRIMARY KEY,
70      exam_type_id INT FOREIGN KEY REFERENCES tbl_exam_type(exam_type_id),
71      name VARCHAR(45),
72      start_date DATE
73      );
74
75      -- attendance table
76      CREATE TABLE tbl_attendance (
77      date DATE,
78      student_id INT FOREIGN KEY REFERENCES tbl_student(student_id),
79      status BIT,
80      remark TEXT
81      );
82
83      -- classroom table
84      CREATE TABLE tbl_classroom (
85      classroom_id INT NOT NULL PRIMARY KEY,
86      year DATE,
87      grade_id INT FOREIGN KEY REFERENCES tbl_grade(grade_id),
88      section CHAR(2),
89      status BIT,
90      remarks VARCHAR(45),
91      teacher_id INT FOREIGN KEY REFERENCES tbl_teacher(teacher_id)
92      );
93
94

```

```

95      -- classroom_student table
96      CREATE TABLE tbl_classroom_student (
97      classroom_id INT FOREIGN KEY REFERENCES tbl_classroom(classroom_id),
98      student_id INT FOREIGN KEY REFERENCES tbl_student(student_id)
99      );
100
101      -- course table
102      CREATE TABLE tbl_course (
103      course_id INT NOT NULL PRIMARY KEY,
104      name VARCHAR(45),
105      description VARCHAR(45),
106      grade_id INT FOREIGN KEY REFERENCES tbl_grade(grade_id)
107      );
108
109      -- exam_result table
110      CREATE TABLE tbl_exam_result (
111      exam_id INT FOREIGN KEY REFERENCES tbl_exam(exam_id),
112      student_id INT FOREIGN KEY REFERENCES tbl_student(student_id),
113      course_id INT FOREIGN KEY REFERENCES tbl_course(course_id),
114      marks INT
115      );

```

## Database Queries:

```

1      -- query1: show all student details from table tbl_student
2      -- ordered by student_id showing the parent full name instead
3      -- of parent_id
4
5      SELECT student_id, email, password, (fname + ' ' + lname)
6      [Student Name], dob, phone, mobile, (SELECT (fname + ' ' + lname)
7      FROM tbl_parent
8      WHERE tbl_student.parent_id=tbl_parent.parent_id) [Parent Name],
9      date_of_join,
10     status, last_login_date, last_login_ip
11     FROM tbl_student
12     ORDER BY student_id ASC;
13
14
15     -- query2: show active student's (status = 1) full name
16     -- and show all his courses and its all exam type marks
17
18     SELECT stdnt.student_id, (stdnt.fname + ' ' + stdnt.lname)
19     [Student Name], course.name [Course Name], ex.name [Exam Name],
20     ex_type.name [Exam Type], ex_res.marks [Exam Marks]
21     FROM tbl_student stdnt JOIN tbl_exam_result ex_res ON
22     stdnt.student_id = ex_res.student_id JOIN tbl_course course ON
23     ex_res.course_id = course.course_id JOIN tbl_exam ex ON
24     ex_res.exam_id = ex.exam_id JOIN tbl_exam_type ex_type ON
25     ex_type.exam_type_id = ex.exam_type_id
26     WHERE stdnt.status = 1;
27
28

```

```

29      -- query3: show every course and its active teacher (status = 1)
30
31      SELECT course.name [Course Name], (tchr.fname + ' ' + tchr.lname)
32      [Teacher Name]
33      FROM tbl_teacher tchr JOIN tbl_classroom clsrom ON
34      tchr.teacher_id = clsrom.teacher_id JOIN tbl_grade grade ON
35      clsrom.grade_id = grade.grade_id JOIN tbl_course course ON
36      course.grade_id = grade.grade_id
37      WHERE tchr.status = 1
38      ORDER BY course.name ASC;
39
40
41      -- query4: for each course, get its average exams types marks
42
43      SELECT course.name, ex.exam_type_id [Exam Type ID],
44      ex.name [Exam Name], AVG(ex_res.marks) [Avg. Marks / Exam Type]
45      FROM tbl_exam_result ex_res JOIN tbl_exam ex ON
46      ex_res.exam_id = ex.exam_id JOIN tbl_course course ON
47      course.course_id = ex_res.course_id
48      WHERE ex_res.course_id = course.course_id
49      GROUP BY course.name, ex.exam_id, ex.exam_type_id, ex.name;
50
51
52      -- query5: show number of students in each classroom arranged
53      -- by classroom_id
54
55      SELECT classroom_id [Classroom ID], AVG(student_id)
56      [No. of Students]
57      FROM tbl_classroom_student
58      GROUP BY classroom_id
59      ORDER BY classroom_id;
60
61
62      -- query6: show all students' attendance details
63
64      SELECT stdnt.student_id, (stdnt.fname + ' ' + stdnt.lname)
65      [Student Name], att.status [Status], att.date [Date]
66      FROM tbl_student stdnt JOIN tbl_attendance att ON
67      att.student_id = stdnt.student_id;
68
69
70      -- query7: show all students and all their parents details
71
72      SELECT stdnt.student_id, (stdnt.fname + ' ' + stdnt.lname)
73      [Student Name], parent.*
74      FROM tbl_student stdnt JOIN tbl_parent [parent] ON
75      stdnt.parent_id = parent.parent_id;
76
77
78      -- query8: show all teachers email ,phones and mobiles
79
80      SELECT (fname + ' ' + lname) [Teacher Name],
81      email [E-Mail], phone [Phone], mobile [Mobile]
82      FROM tbl_teacher
83      ORDER BY fname;
84
85

```

```

86      -- query9: show active students that have zero marks in any exam,
87      -- showing the exam name and type
88
89      SELECT stdnt.student_id [Student ID],
90      (stdnt.fname + ' ' + stdnt.lname) [Name], ex_type.name [Exam Type],
91      ex.exam_id [Exam ID], ex.name [Exam Name]
92      FROM tbl_student stdnt JOIN tbl_exam_result ex_res ON
93      stdnt.student_id = ex_res.student_id JOIN tbl_exam ex ON
94      ex_res.exam_id = ex.exam_id JOIN tbl_exam_type ex_type ON
95      ex_type.exam_type_id = ex.exam_type_id
96      WHERE stdnt.status = 1 AND (ex_res.marks = 0 or ex_res.marks = NULL)
97      ORDER BY stdnt.student_id;
98
99
100     -- query10: get email and phone of parents of students that
101     -- didn't attend for more than a month
102
103     SELECT stdnt.student_id [Student ID],
104     (stdnt.fname + ' ' + stdnt.lname) [Student Name], MAX(att.date)
105     [Last Attendance], (parent.fname + ' ' + parent.lname)
106     [Parent Name], parent.email [E-Mail], parent.phone [Phone],
107     parent.mobile [Mobile]
108     FROM tbl_student stdnt JOIN tbl_attendance att ON
109     att.student_id = stdnt.student_id JOIN tbl_parent [parent] ON
110     stdnt.parent_id = parent.parent_id
111     WHERE att.status = 0
112     AND att.date <= DATEADD(month, -1, CONVERT(date, getdate()))
113     GROUP BY stdnt.student_id, (stdnt.fname + ' ' + stdnt.lname),
114     (parent.fname + ' ' + parent.lname),
115     parent.email, parent.phone, parent.mobile;
116
117
118     -- query11: get courses names for grades greater than 3 ordered
119     -- alphabetically ascended
120
121     SELECT * FROM tbl_course
122     WHERE grade_id > 3
123     ORDER BY tbl_course.name ASC;
124
125
126     -- query12: order students names alphabetically and show their
127     -- details
128
129     SELECT * FROM tbl_student
130     ORDER BY fname ASC;
131
132
133     -- query13: show the maximum mark in each exam type and the student
134     -- who got that mark and in what course
135
136     SELECT (stdnt.fname + ' ' + stdnt.lname) [Student Name],
137     course.name [Course Name], ex.name [Exam Name], ex_type.name
138     [Exam Type], MAX(ex_res.marks) [Max Marks]
139     FROM tbl_student stdnt JOIN tbl_exam_result ex_res ON
140     stdnt.student_id = ex_res.student_id JOIN tbl_course course ON
141     ex_res.course_id = course.course_id JOIN tbl_exam ex ON
142     ex_res.exam_id = ex.exam_id JOIN tbl_exam_type ex_type ON
143     ex_type.exam_type_id = ex.exam_type_id
144     GROUP BY (stdnt.fname + ' ' + stdnt.lname), course.name,
145     ex.name, ex_type.name
146     ORDER BY course.name;
147
148

```

```
149      -- query14: show all students that are 8+ years old
150
151      SELECT (fname + ' ' + lname) [Student], dob [Date of Birth]
152      FROM tbl_student
153      WHERE dob <= DATEADD(year, -8 , CONVERT(date, getdate()));
154
155
156      -- query15: show all teachers that are 40-60 years old
157
158      SELECT (fname + ' ' + lname) [Teacher], dob [Date of Birth]
159      FROM tbl_teacher
160      WHERE dob BETWEEN DATEADD(year, -40 , CONVERT(date, getdate()))
161      AND DATEADD(year, -60 , CONVERT(date, getdate()));
```